

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Petra Veselá



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**SOFTWAREVÝ GENERÁTOR TESTOVACÍCH SIGNÁLŮ
PRO AKUSTICKÁ MĚŘENÍ**

SOFTWARE GENERATOR OF TEST SIGNALS FOR ACOUSTIC MEASUREMENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Petra Veselá

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Frenštátský

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Audio inženýrství**

Ústav telekomunikací

Studentka: Petra Veselá

ID: 164625

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Softwarový generátor testovacích signálů pro akustická měření

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte algoritmy pro generování testovacích signálů pro akustická měření, zejména algoritmus pro generování barevných šumů (bílý, růžový, hnědý, modrý a fialový) umožňující řízení činitele výkyvu. Dalšími generovanými signály bude binární šum, signál složený z harmonických složek s danými parametry a lineárně a logaritmicke přeladovaný harmonický signál. Navržené algoritmy implementujte do zásuvného modulu s využitím technologie VST2.

DOPORUČENÁ LITERATURA:

[1] SYSEL, P.; SMÉKAL, Z. Číslicové filtry. Brno: Vysoké učení technické v Brně, 2012. s 1-148. ISBN 978-8-214-4454-6

[2] MÜLLER, G.; MÖSER, M. Handbook of Engineering Acoustics. Springer-Verlag Berlin Heidelberg, 2013. 706s. ISBN 978-3-540-69460-1

Termín zadání: 1.2.2016

Termín odevzdání: 1.6.2016

Vedoucí práce: Ing. Petr Frenštátský

Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem předkládané bakalářské práce je vytvoření zásuvného modulu technologie VST2, který bude schopen generovat testovací signály použitelné při akustických měřeních. Součástí práce je teoretický rozbor vlastností jednotlivých signálů a možností jejich využití při samotném měření, dále také stručný popis technologie VST a použitých knihoven. Závěrečná část pak přináší zhodnocení parametrů výsledných signálů.

KLÍČOVÁ SLOVA

barevné šumy, činitel výkyvu, impulsní odezva, signál složený z harmonických složek, posloupnost maximální délky, přeladovaný harmonický signál, VST2

ABSTRACT

The aim of the thesis is to create a VST2 plug-in module capable of generating input signals for different kinds of acoustic measurements. The opening section offers a theoretical overview of various test signals, their parameters and possibilities of their use during measurement. Subsequent divisions are then introducing a description of several methods of acoustic measurements and a brief overview of a VST interface. Last chapter offers the analysis of the created signals and discussion of their potential use in practice.

KEYWORDS

colored noise, crest factor, frequency sweep, impulse response, Maximum Length Sequence, multitone, VST2

VESELÁ, Petra *Softwarový generátor testovacích signálů pro akustická měření*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 47 s. Vedoucí práce byl Ing. Petr Frenštátský

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Softwarový generátor testovacích signálů pro akustická měření“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autorky

PODĚKOVÁNÍ

Na tomto místě chci poděkovat vedoucímu své bakalářské práce Ing. Petru Frenštátskému za odborné vedení, konzultace, velkou trpělivost a podnětné návrhy k práci. Dále bych také ráda poděkovala Bc. Jindřichu Pevnému, zejména za pomoc a rady při implementaci.

Brno

.....

podpis autorky

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
podpis autorky

OBSAH

1	Úvod	11
2	Testovací signály	12
2.1	Náhodný a pseudonáhodný šum	12
2.2	Činitel výkyvu	12
2.3	Širokopásmové šumy	12
2.3.1	Bílý šum	13
2.3.2	Růžový šum	15
2.3.3	Hnědý šum	15
2.3.4	Modrý a fialový šum	16
2.4	Pseudonáhodný binární šum	16
2.5	Přeladovaný harmonický signál	19
2.6	Signál složený z harmonických složek	20
3	Metody akustických měření	22
3.1	Rychlá Fourierova transformace	22
3.2	Time Delay Spectrometry	22
3.3	<i>FASTTEST</i>	23
4	Implementace	25
4.1	VST - Virtual Studio Technology	25
4.2	FFTW	25
5	Výsledky práce	27
5.1	Struktura programu	27
5.2	Spuštění a ovládání aplikace	28
5.3	Rozbor výsledných signálů	31
5.3.1	Bílý šum	31
5.3.2	Barevné šumy	31
5.3.3	Pseudonáhodný binární šum	32
5.3.4	Přeladovaný harmonický signál	33
5.3.5	Signál složený z harmonických složek	33
6	Závěr	34
	Literatura	35
	Seznam symbolů, veličin a zkratk	37

Seznam příloh	38
A Spektrální vlastnosti výsledných signálů	39
B Ukázky kódu	45
C Obsah přiloženého DVD	47

SEZNAM OBRÁZKŮ

2.1	Průběhy mapovací funkce pro $n = 1$, $n = 2$ a $n = 3$	14
2.2	Algoritmus generování MLS.	16
3.1	Schéma zpracování signálu při měření technikou TDS.	23
5.1	Grafické uživatelské rozhraní pluginu.	30
5.2	Algoritmus generování barevných šumů.	32
A.1	Autokorelační funkce MLS s délkou masky 16 bitů.	39
A.2	PSD bílého šumu s různými hodnotami CF.	40
A.3	Spektrální vlastnosti růžového a červeného šumu.	41
A.4	Spektrální vlastnosti modrého a fialového šumu.	42
A.5	Spektrální vlastnosti přeladovaných harmonických signálů.	43
A.6	Spektrální vlastnosti signálů složených z harmonických složek.	44

SEZNAM TABULEK

2.1	Primitivní polynomy, převzato z [3].	17
3.1	Střední kmitočty třetinooktávových pásem, převzato z [17].	24

1 ÚVOD

Většina akustických měření je založena na principu buzení zkoumaného systému signálem známých vlastností a měření odezvy na jeho výstupu. Pro získání relevantních výsledků je zásadní použít vhodný budící signál příslušných vlastností. Běžně používanými testovacími signály jsou například bílé a barevné šумы, pseudonáhodný binární šum, přeladovaný harmonický tón nebo signál složený z harmonických složek, tzv. *multitone*.

Cílem této práce je implementovat algoritmy pro generování výše zmíněných signálů, který byly navrženy v předcházejícím semestrálním projektu v prostředí MATLAB, do zásuvného modulu technologie VST2 pomocí jazyka C++ a ověřit jeho funkčnost. Výsledky předkládané práce by měly být dále použity při vývoji aplikace *Audio FX Analyzer*.

2 TESTOVACÍ SIGNÁLY

2.1 Náhodný a pseudonáhodný šum

První skupinu generovaných signálů tvoří šumy. Náhodným šumem rozumíme takovou posloupnost amplitud v časové oblasti, která je neperiodická a nedeterministická. Pseudonáhodný šum se při pozorování v omezeném časovém rozsahu může jevit jako náhodný, nicméně při dostatečně dlouhém pozorování lze určit v jeho průběhu periodu. Oproti náhodnému signálu je pseudonáhodný signál deterministický a tedy i „opakovatelný“. To může být výhodné, protože měření systému tak lze provést několikrát, za stejných podmínek.

Lze ukázat, že ve spektrální oblasti při dostatečně dlouhém pozorování obsahuje skutečně náhodný šum energii na všech frekvencích. Pseudonáhodný šum naproti tomu má spektrum diskrétní a energii nesou pouze složky odpovídající reciproké hodnotě periody v časové oblasti a jejich celočíselných násobků [12].

2.2 Činitel výkyvu

Jedním z nejdůležitějších parametrů testovacích signálů je činitel výkyvu neboli CF (crest factor), který je definován jako poměr mezi maximální hodnotou signálu a jeho efektivní hodnotou. Popisuje tak variabilitu signálu – přítomnost impulsů v něm (například harmonický signál má CF roven $\sqrt{2}$, tedy nízký, naproti tomu ideální šum s gaussovým rozložením pravděpodobnosti má CF teoreticky nekonečný). Při audio měřeních bývá tento poměr většinou udáván v dB, např. viz [16]. V případě náhodného ergodického šumu pak platí pro CF tento vztah [19]:

$$CF = 20 \cdot \log_{10} \frac{x_{max}}{\sigma_x}. \quad (2.1)$$

Obecně platí, že většina audio měření požaduje testovací signál s nízkým CF [2], který při měření vykazuje nižší energetické ztráty a je tak schopný lépe vybudit měřenou soustavu. Tuto vlastnost lépe než barevné šumy splňují zejména pseudonáhodný binární šum a přeladovaný harmonický signál.

2.3 Širokopásmové šumy

Pod pojmem šum obvykle v oblasti audio inženýrství rozumíme náhodný signál nízké úrovně, který tvoří parazitní složku užitečného signálu. Šumy však také tvoří velkou skupinu testovacích signálů používaných při audio měřeních. Širokopásmové šumy se běžně používají ke zjišťování frekvenční odezvy systému. Prvoplánový postup při

měření impulsní odezvy je, že na vstup systému přivedeme signál podobný Diracovu impulsu a na výstupu zaznamenáme přímý výsledek. V praxi lze tento teoretický signál aproximovat například výstřelem z testovací pistole. Měření pomocí šumů však umožňuje dosáhnout podstatně lepšího odstupu užitečného signálu od šumu (SNR). V oblasti číslicového zpracování signálu jsou šумы reprezentovány posloupnostmi náhodných, resp. pseudonáhodných čísel.

2.3.1 Bílý šum

Bílý šum je charakteristický svojí konstantní PSD (Power Spectral Density, výkonová spektrální hustota¹), pro potřeby audio měření postačí aby byla konstantní ve slyšitelném rozsahu (20 Hz – 20 kHz). To znamená, že signál má v každém frekvenčním pásmu stejné absolutní šířky stejný výkon. Této vlastnosti se využívá zejména při zjišťování frekvenční odezvy audio zařízení.

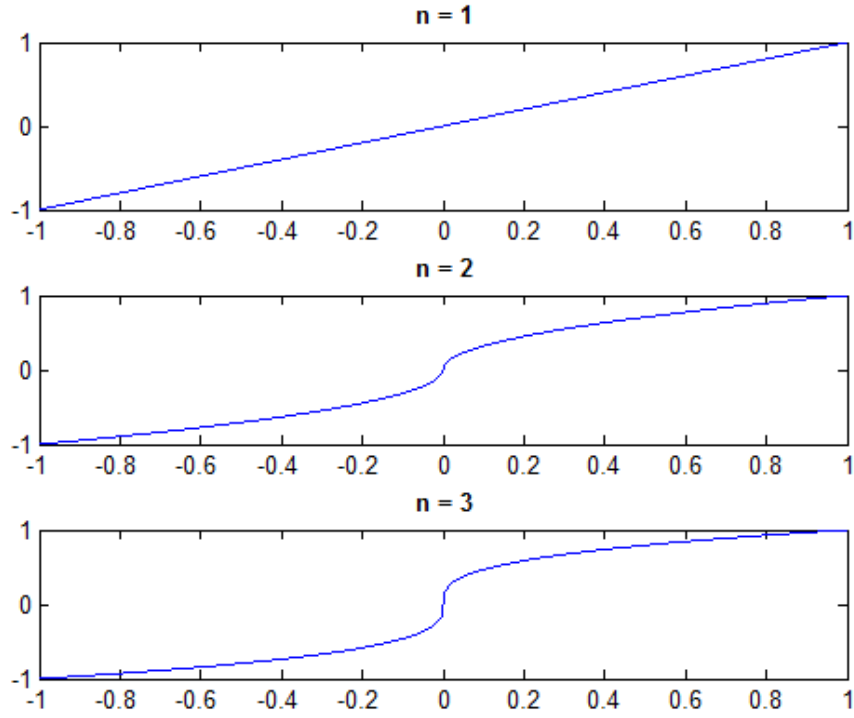
V analogové technice se bílému šumu blíží teplotní šum aktivních součástek. Takový šum má přibližně Gaussovo rozložení pravděpodobnosti, v digitální oblasti, například v prostředí MATLAB, ho lze optimálně simulovat jako vektor vygenerovaný funkcí `randn()`, založené na algoritmu Zikkurat [14], která vrací pseudonáhodná čísla s gaussovým rozložením.

Nicméně, požadavkem zadání byla možnost parametrického zadávání hodnoty činitele výkyvu v dB a algoritmus navržený v [10] požaduje za vstup vektor pseudonáhodných čísel s uniformním rozdělením. Z tohoto důvodu byla ve výsledném programu použita funkce knihovny `cstdlib`, `rand()`. Hodnoty vektoru jsou dále normovány na rozsah $\langle -1; 1 \rangle$. Následně je na vektor aplikována mapovací funkce, která modifikuje původní pravděpodobnostní funkci a pozmění tak výsledný CF na hodnotu přibližně rovnou hodnotě požadované (s přesností na několik desetin dB).

Podle vztahu 2.1 závisí hodnota CF na maximální výchylce signálu a jeho efektivní hodnotě. Je-li vektor vygenerovaný funkcí `rand()` normován tak, že absolutní hodnota maxima vektoru je vždy rovna 1, pak hodnota CF závisí pouze na RMS. Hodnota CF před zlogaritmováním odpovídá $\frac{1}{RMS}$. Proměnná n je pomocný parametr k určení tvaru mapovací funkce tak, aby byl vytvořen vektor s požadovanou RMS.

Mapovací funkce $y(x)$ (2.2) bere za parametr postupně všechny vzorky vektoru \mathbf{x} s uniformním rozdělením a hodnoty amplitud přemapuje tak, aby výsledná hodnota

¹[20], str. 203



Obr. 2.1: Průběhy mapovací funkce pro $n = 1$, $n = 2$ a $n = 3$.

RMS odpovídala právě reciproké hodnotě požadovaného CF.

$$y(x) = \begin{cases} e^{\frac{\ln(x)}{n}}, & x > 0 \\ 0, & x = 0 \\ -1 \cdot e^{\frac{\ln(x)}{n}}, & x < 0 \end{cases} \quad (2.2)$$

Vztah mezi parametrem n a RMS hodnotou výsledného signálu lze odvodit substitucí. Vektor prvků s uniformním rozdělením odpovídá funkci $x(t)$, jeho efektivní hodnotu lze spočítat ze vztahu:

$$RMS_x = \sqrt{\int_0^1 x^2(t) dt} \quad (2.3)$$

Po zpracování vektoru mapovací funkcí (a tedy substitucí $y(x) = x(t)$) lze psát:

$$RMS_x = \sqrt{\int_0^1 e^{2 \cdot \frac{\ln(x)}{n}} dx} \quad (2.4)$$

Integrací získáme vztah:

$$RMS_x = \sqrt{\left[\frac{n \cdot x^{\frac{n+2}{n}}}{n+2} \right]_0^1} = \sqrt{\frac{n}{n+2}},$$

z něhož lze vyjádřit závislost hodnoty n na RMS signálu:

$$n = \frac{2 \cdot RMS^2}{1 - RMS^2} \quad (2.5)$$

Obrázek 2.1 zobrazuje příklad průběhu mapovací funkce pro $n = 1, 2, 3$

Podle [8] není možné tímto algoritmem vygenerovat šum s jiným než vyrovnaným frekvenčním spektrem. Lze však touto metodou vygenerovat nejprve vektor bílého šumu s daným CF a z něj pak filtrací vytvořit barevný šum. Takto získaný barevný šum sice nebude mít stejný CF jako prve vygenerovaný bílý šum, nicméně při použití bílého šumu s vysokým CF je pravděpodobné, že získaný barevný šum bude mít CF také vysoký.

2.3.2 Růžový šum

PSD růžového šumu má směrnici -3 dB/okt. (resp. -10 dB/dekádu), při třetino-oktávové analýze má však frekvenční spektrum konstantní. Oktávovou resp. třetino-oktávovou analýzu využívají zvukoměry při zjišťování celkové hlučnosti signálu. Slyšitelné spektrum je rozděleno na oktávy, resp. třetiny oktáv a je zjišťován výkon v těchto jednotlivých pásmech s procentuálně konstantní šířkou. Pro určení celkové hlučnosti totiž není přesná znalost spekter nutná. Růžový šum se používá zejména k měření frekvenčních odezev poslechových místností nebo koncertních sálů, protože kumuluje na nízkých kmitočtech více energie a ze všech barevných šumů se tak nejvíce přibližuje logaritmickému charakteru lidského sluchu, který je na nízkých kmitočtech méně citlivý. Pro jeho generování byl v práci použit algoritmus založený na principu rychlé konvoluce.

Nejprve je vygenerován vektor vzorků bílého šumu se zadaným CF (ten je však pro výsledný barevný šum spíše orientační).

Signál je dále rychlou Fourierovou transformací převeden do frekvenční oblasti. Zde je pak souměrné modulové spektrum příslušně váhováno funkcí, která odpovídá odmocnině požadované obálky PSD – například pro růžový šum, jehož obálka PSD je nepřímo úměrná první mocnině frekvence ($1/f$), se spektrum násobí s vektorem odpovídajícím sklonu $1/\sqrt{f}$, v případě hnědého šumu má váhovací vektor charakteristiku $1/f$ atd. Tento způsob modelování modulového spektra je sice výpočetně náročnější než použití číslicových filtrů, odpadá však problém s jejich návrhem.

2.3.3 Hnědý šum

Dalším implementovaným šumem je hnědý šum, jehož PSD má směrnici -6 dB/okt. Přesnějším označením je tedy spíše „červený šum“, které odkazuje na paralelu s viditelným světlem, kde červené světlo má nejnižší frekvenci z celého viditelného spektra.

Hnědý šum je tak spíše nepřesným překladem anglického „Brownian noise“, popisujícím náhodný Brownův pohyb molekul – dalším synonymem tohoto šumu v anglické literatuře je „Random Walk noise“. Hnědý šum se běžně jako vstupní signál pro akustická měření nepoužívá, své uplatnění však nachází při konstrukci reproboxů, kdy se pomocí něj nastavuje fáze subwooferu v crossoveru s dalšími reproduktory v soustavě.

2.3.4 Modrý a fialový šum

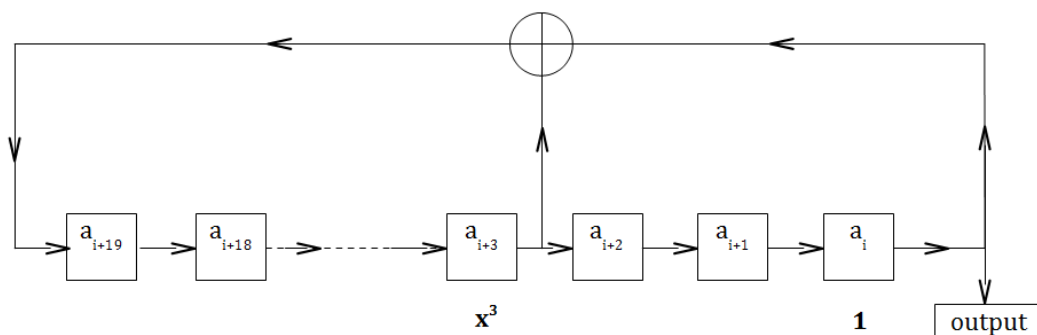
Ani modrý (směrnice PSD +3 dB/okt.) a fialový šum (+6 dB/okt.) nebývají používány jako testovací signály, velké využití však mají při ditheringu – přidáním nízké úrovně těchto šumů s vyšším zastoupením vyšších frekvencí lze dosáhnout snížení kvantizační chyby vzniklé při použití nízké bitové hloubky záznamu.

2.4 Pseudonáhodný binární šum

Obdobou bílého šumu s vyrovnanou výkonovou hustotou je v digitální oblasti pseudonáhodný binární šum, konkrétně MLS, tedy posloupnost maximální délky. Jedná se o posloupnost jedniček a nul generovanou posuvným registrem s operací součet modulo dvě zavedenou ve zpětné vazbě. Pro použití při akustickém měření jsou binární stavy 0 a 1 mapovány vztahem 2.6 na +1 resp. −1. Snižuje se tím stejnosměrná složka, což je výhodné z hlediska minimálních energetických ztrát při samotném měření.

$$s_0, s_1, s_2, s_3 \dots = (-1)^{a_0}, (-1)^{a_1}, (-1)^{a_2}, (-1)^{a_3} \dots \quad (2.6)$$

Blokové schéma generátoru je uvedeno na obrázku 2.2.



Obr. 2.2: Algoritmus generování MLS.

Aby byla posloupnost skutečně nejdelší možná a obsahovala tak všechny možné binární kombinace (s výjimkou nulového registru – ten generuje vždy jen posloupnost nul), které lze pro registr dané délky získat, je nutné použít pro rekurzivní vztah takovou masku, která má jedničky na bitech odpovídajících nenulovým koeficientům zvoleného primitivního polynomu příslušného řádu.

Pro registry delší než 4 bity existuje více primitivních polynomů a tedy více různých rekurzivních vztahů. Podle [6] jsou však spektrální vlastnosti signálů vygenerovaných těmito variantami rovnocenné. V bakalářské práci proto implementuji pro každou z délek 16 až 21 bitů jedinou masku. Například 20bitová maska 00000000000000001001, která odpovídá primitivnímu polynomu $x^{20} + x^3 + 1$, má zpětnovazební „taps“ („kohoutky“) na bitech s indexem 16 a 19. Signál vygenerovaný pomocí této masky obsahuje $M = 2^{20} - 1 = 1048575$ vzorků, což při přehrávání se vzorkovací frekvencí 44,1 kHz znamená téměř 24 s testovacího signálu. Další polynomy jsou uvedeny v tabulce 2.1.

Tab. 2.1: Primitivní polynomy, převzato z [3].

Délka registru	Příklad primitivního polynomu	Délka testovacího signálu při $f_{vz} = 44,1$ kHz [s]
16	$x^{16} + x^{12} + x^3 + x + 1$	1,48
17	$x^{17} + x^3 + 1$	2,97
18	$x^{18} + x^7 + 1$	5,94
19	$x^{19} + x^5 + x^2 + x + 1$	11,9
20	$x^{20} + x^3 + 1$	23,7
21	$x^{21} + x^2 + 1$	47,6

Není podstatné, jaký výchozí vstup zvolíme (s výjimkou nulového registru – ten by generoval nulový signál), protože registr postupně prochází všemi možnými bitovými kombinacemi, proto přehlednost volím vektor jedniček.

MLS mají tři významné vlastnosti [3]:

a) Balanční kritérium (Balance Property)

Každá MLS obsahuje právě 2^{n-1} jedniček a právě $2^{n-1} - 1$ nul, kde n je počet bitů použitého registru.

b) Běhové kritérium (Run Property)

Každá MLS obsahuje právě 2^{n-1} „běhů“, tedy po sobě jdoucích jedniček, nebo nul. Právě $1/2^n$ všech běhů je délky n bitů s tím, že právě 1 běh je délky n bitů a právě 1 běh je délky $n - 1$ bitů.

c) Autokorelační vlastnost (Correlation Property)

Autokorelace konečné posloupnosti reálných členů je definována vztahem

$$r_{xx}[k] = \sum_{n=-\infty}^{+\infty} x[n+k] \cdot x[n], \quad k = 0, \pm 1, \pm 2. \quad (2.7)$$

Normalizovaná autokorelační funkce MLS je vyjádřena jako součet periodické posloupnosti Diracových impulsů (periodou je délka signálu M) a stejnosměrné složky $-\frac{1}{M}$, která je při dostatečných délkách M zanedbatelná:

$$r_{xx} = \delta(k) - \frac{1}{M}. \quad (2.8)$$

Autokorelační funkce blízká Diracovu impulsu ukazuje na „bílé“ spektrum, MLS lze tedy pro měření použít obdobným způsobem jako bílý šum, ovšem se všemi výhodami deterministického signálu, které jsou popsány dále.

Platí, že křížová korelace vstupního a výstupního signálu odpovídá lineární konvoluci autokorelace vstupního signálu a impulsní odezvy systému:

$$r_{xy}[k] = r_{xx}[k] * h[k] \quad (2.9)$$

Pokud tedy člen $r_{xx}[k]$ odpovídá periodickému sledu Diracových impulsů, lze zjistit impulsní odezvu systému prostou křížovou korelací vstupního a výstupního signálu, protože výsledkem konvoluce sekvence s Diracovým impulsem je opět původní sekvence:

$$h[k] \approx r_{xy}[k]. \quad (2.10)$$

V praxi se při měření pomocí MLS používá místo jinak běžné rychlé Fourierovy transformace (FFT) rychlá Hadamardova transformace (FHT). Její hlavní předností oproti FFT je skutečnost, že signálu MLS chybí právě jeden vzorek do délky 2^n , pro kterou je použití FFT nejefektivnější - vložení jednoho nulového vzorku by přitom zásadně porušilo vyrovnané frekvenční spektrum. FHT je proto pro zpracování MLS mnohem efektivnější – umožňuje získání impulsní odezvy bez nutnosti převodu signálu do frekvenční oblasti ².

²[5], str. 99.

Teoreticky signál MLS dosahuje velmi nízkých hodnot CF, ideálně až 0 dB. V praxi však v důsledku zkreslení při DA převodu softwarově generovaného signálu dochází k podstatnému zkreslení – aby mu bylo zabráněno, musí být použit signál s úrovní alespoň 5–8 dB pod full scale úrovní. Výsledný signál pak bude mít o to nižší energii a na výstupu tak bude dosaženo nižších hodnot SNR.

Navíc jsou výsledky měření pomocí MLS poměrně náchylné k chybám způsobených časovou variantností měřeného systému. Z hlediska této a několika dalších vlastností je proto výhodné použít jako testovací signál pro zjišťování impulsní odezvy přeladovaný harmonický signál [11].

2.5 Přeladovaný harmonický signál

Pro tento signál se i v české literatuře lze často setkat s anglickými názvy *sweep* nebo *chirp*. Jedná se o sinový nebo kosinový signál, jehož okamžitá frekvence $f(t)$ roste v určitém intervalu frekvencí buď lineárně, nebo logaritmicky.

Lineárně přeladovaný sweep má spektrum bílé, logaritmicky přeladovaný růžové (sklon PSD -3 dB/okt., s výjimkou nejnižších frekvencí).

Přeladovaný signál je možné generovat jak v časové, tak i ve frekvenční oblasti, přičemž každá možnost má své výhody a nevýhody [11]. Signál přímo syntetizovaný v časové oblasti má dokonalou časovou obálku a CF přesně roven činiteli výkyvu sinové vlny (3,02 dB), spektrum však vykazuje určité nerovnosti vzniklé v důsledku náhlého začátku a konce generování. Tyto nedostatky je možné částečně eliminovat použitím okénkové funkce. Generování ve spektrální oblasti spočívá v definování modulové charakteristiky a skupinového zpoždění spektra přeladovaného signálu, převodu prvků spektra do algebraického tvaru a následné zpětné transformaci do časové oblasti. V rámci této bakalářské práce bylo implementováno pouze generování v časové oblasti.

Pro okamžitou frekvenci při lineárním přeladování platí vztah 2.15, při logaritmickém 2.12:

$$f_{lin} = f_0 + k \cdot t, \quad (2.11)$$

$$k = \frac{f_1 - f_0}{T},$$

$$f_{log}(t) = f_0 \cdot k^t, \quad (2.12)$$

$$k = \left(\frac{f_1}{f_0} \right)^{\frac{1}{T}},$$

kde f_0 je počáteční frekvence sweepu, f_1 konečná frekvence sweepu a T je doba, za kterou je signál přeladěn z frekvence f_0 na f_1 .

Pro generování signálu je však nutné znát okamžitou fázi signálu, resp. argument harmonické funkce pro každý generovaný vzorek. Vztah pro okamžitou fázi lze získat integrací vztahů 2.15 a 2.12:

$$\phi(t)_{lin} = 2\pi \cdot \int f_{lin}(t)dt = 2\pi \left(f_0 \cdot t + k \cdot \frac{t^2}{2} \right), \quad (2.13)$$

$$\phi(t)_{log} = 2\pi \cdot \int f_{log}(t)dt = 2\pi \cdot f_0 \cdot \frac{k^t}{\ln(k)}. \quad (2.14)$$

Prostou implementací těchto rovnic získáme přeladovaný harmonický signál.

Oproti měření impulsní odezvy pomocí MLS má měření se sweepem lepší imunitu proti harmonickému zkreslení – produkty harmonického zkreslení mohou být z výsledné impulsní odezvy snadno izolovány³ Dále jsou také tato měření dobře odolná vůči časové variantnosti měřeného systému, proto se dobře hodí ke zjištění odezev analogových zařízení nebo při outdoorových měřeních.

2.6 Signál složený z harmonických složek

Jak už naznačuje název tohoto signálu (angl. *multitone*), jedná se o součet několika harmonických složek. Každá z těchto složek je určena právě třemi parametry (amplituda A , frekvence f a počáteční fáze ϕ) a jejich počet závisí na daném typu akustického měření. Častou konfigurací je například použití dvou harmonických složek při měření intermodulačního zkreslení (podíl intermodulačních složek $(m \cdot f_1 \pm n \cdot f_2)$ na výstupním výkonu). Standard CCIF doporučuje použít dvě složky se stejnou amplitudou a blízkými kmitočty (konkrétně 19 + 20 kHz při měření systému s plným pokrytím slyšitelného pásma). V praxi jsou však častěji používány standardy SMPTE/DIN, které jako testovací signály používají lineární kombinaci nízkofrekvenční složky s amplitudou čtyřnásobnou oproti kmitočtu druhé, vysokofrekvenční složky. SMPTE doporučuje konkrétně 60 Hz + 7 kHz, DIN potom 250 Hz + 8 kHz[12].

Další možností využití tohoto signálu je měření frekvenční odezvy systému. V tomto případě se používají multitóny obsahující průměrně 10–30 složek. Zřejmou výhodou oproti přeladovanému harmonickému signálu je časová úspora při měření, další možnosti jsou popsány v následujících kapitolách.

Pro generování jednotlivých harmonických byl v programu zvolen jednoduchý algoritmus fázového akumulátoru, kdy pro každý nově generovaný vzorek je fáze

³[11], str. 10

cyklicky inkrementována o konstantní hodnotu závislou na požadované frekvenci signálu a použité vzorkovací frekvenci:

$$incr_{ph} = 2 \cdot \pi \cdot \frac{f}{f_{vz}}. \quad (2.15)$$

Tento cyklus je volán pro každou jednu harmonickou složku a výsledky se kumulativně sčítají ve vnitřním bufferu. Po vygenerování všech složek s nenulovou amplitudou je vektor výsledků normován na standardní rozsah hodnot $\langle -1; +1 \rangle$.

3 METODY AKUSTICKÝCH MĚŘENÍ

3.1 Rychlá Fourierova transformace

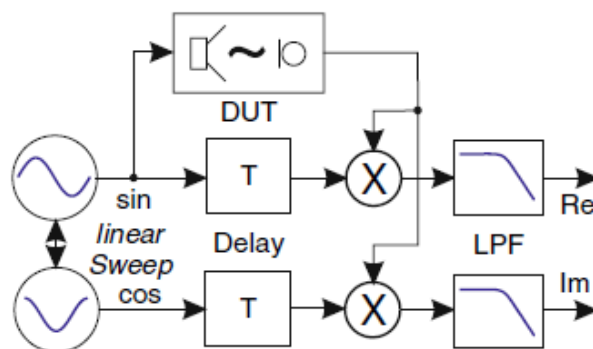
Pod zkratkou FFT rozumíme rychlou Fourierovu transformaci, tedy převod z časové do frekvenční oblasti pomocí efektivního algoritmu, který byl představen v roce 1965 v [4]. Algoritmus FFT využívá pouze základní algebraické operace – sčítání a násobení. Jeho princip spočívá ve skutečnosti, že funkce e^{jx} je periodická a některé členy výsledné posloupnosti tak lze sdružovat.

Nejvyšší efektivity dosahuje algoritmus FFT při aplikaci na signály o délce 2^n vzorků. Výsledkem FFT je diskrétní množina dat závislosti amplitudy na frekvenci s polovičním počtem prvků oproti vstupním datům amplitudově-časové závislosti. Tyto jednotlivé prvky se nazývají „biny“ a odpovídají lineárně rozmístěným frekvencím v intervalu od 0 Hz (hodnota v nulovém indexu odpovídá stejnosměrné složce) do $f_{vz}/2$ (tzv. Nyquistova frekvence). Pro správnou spektrální analýzu signálu je někdy nutné před provedením FFT vynásobit vstupní signál se speciální okénkovou funkcí, která může zmírnit diskontinuity v okrajových vzorcích transformované sekvence. Pro signály s různým frekvenčním obsahem se hodí různé okénkové funkce, mezi nejčastěji používané patří Hannovo okno (vhodné zejména pro analýzu signálu složeného z harmonických složek). Není-li použita žádná speciální okénková funkce, lze hovořit o aplikaci pravoúhlé okénkové posloupnosti, která vybere z teoreticky nekonečného vstupního signálu konečný úsek (např. u analýzy širokospektrálních šumů).

3.2 Time Delay Spectrometry

Jednou ze základních metod pro zjištění impulsní odezvy je technika TDS, Time Delay Spectrometry. Poprvé ji ve článku [9] představil Richard Heyser v šedesátých letech minulého století. Měření touto metodou je možné provádět čistě analogovou cestou, což je jeden z hlavních důvodů její dlouholeté popularity nastalé ještě před nástupem digitální techniky umožňující FFT. Testovacím signálem je lineárně přeladovaný harmonický signál a oproti měření s MLS tato metoda vyniká zejména potlačením korelovaného i nekorelovaného šumu, ale i imunitou vůči harmonickému zkreslení vzniklému v důsledku nelinearity měřeného systému (například reproduktoru).

Základní schéma zpracování signálu při TDS je znázorněno na blokovém schématu 3.1 převzatém z [5].



Obr. 3.1: Schéma zpracování signálu při měření technikou TDS.

První částí TDS analyzátoru je generátor produkující současně lineárně přeladovaný sinusový signál a s ním fázově synchronizovaný kosinový přeladovaný signál. Testovacím signálem přiváděným na vstup měřeného systému je přeladovaný sinus a jeho podoba na výstupu je pak násobena zaprvé s původním signálem – tím získáme reálnou část přenosové funkce – a zadruhé s kosinovým signálem, čímž získáme její imaginární část. Výstupy z násobiček (které fungují na podobném principu jako směšovací obvod v superhetu) jsou posléze zpracovány filtry typu dolní propusti s pevně nastaveným mezním kmitočtem, které odfiltrují součtové frekvence.

Vzhledem k tomu, že zvuk procházející měřenou soustavou dorazí k měřicímu mikrofону na výstupu se zpožděním, bude jeho okamžitá frekvence nižší než frekvence signálu vycházejícího přímo z generátoru. Z tohoto důvodu je ještě před násobičkami zařazen do obvodu zpožďující blok[5].

3.3 *FASTTEST*

Výraz *FASTTEST* je registrovaná značka společnosti Audio Precision označující implementaci určitých vylepšení základního konceptu měření pomocí multitónu. Při volbě kmitočtů jednotlivých harmonických složek jako celočíselných násobků rozlišovací schopnosti FFT¹ není nutné v analyzátoru používat okénkovou funkci. Signál je zaznamenán po průchodu systémem, je provedena FFT a analýzou různých sad binů lze posléze získat informace o různých parametrech systému (frekvenční odezva, THD, u stereo zařízení mezikanálový fázový posun atd.). Podrobnosti k této metodě lze nalézt například v [12].

Testovacím signálem pro tento typ měření je multitón složený obvykle ze 3–30 harmonických složek, logaritmicky rovnoměrně rozmístěných po slyšitelném frek-

¹[12], str.75

venčním pásmu. Frekvence bývají tedy často voleny jako střední kmitočty třetino-oktávových pásem, které znázorňuje tabulka 3.1.

Tab. 3.1: Střední kmitočty třetino-oktávových pásem, převzato z [17].

	25	50	100	200	400	800	1600	3150	6300	12500
$f [Hz]$	31,5	63	125	250	500	1000	2000	4000	8000	16000
	40	80	160	315	630	1250	2500	5000	10000	20000

Jak už název napovídá, hlavní výhodou této techniky je její rychlost - doba trvání měření v určitých konfiguracích může klesnout až k 270 ms[12]. V některých situacích může být také výhodné, že multitón může být změnou amplitud jednotlivých složek spektrálně tvarován tak, aby vyhovoval právě měřenému systému.

4 IMPLEMENTACE

4.1 VST - Virtual Studio Technology

Generátor výše popsaných testovacích signálů byl implementován do zásuvného modulu technologie VST, kterou vyvíjí společnost Steinberg už od roku 1996[13]. K vývoji vlastních modulů je nutné se bezplatně zaregistrovat na stránkách firmy Steinberg. Po odsouhlasení licenčních podmínek je uživateli umožněno stažení aktuální verze VST SDK. Základem tohoto balíčku je několik zdrojových a hlavičkových souborů v jazyce C++ určených k implementaci processingu, soubory knihovny VSTGUI, příslušnou dokumentaci a příklady implementace jednoduchých funkčních pluginů (včetně předchystaných VC++ projektů pro usnadnění jejich kompilace).

Pro účely této bakalářské práce byla zvolena VST SDK verze 2.4, především z důvodů větší přehlednosti a intuitivnosti oproti aktuální verzi 3.6.5 (ke dni 30.5.2016) a dále také proto, že pro tuto verzi je k dispozici šablona, jejímž autorem je Ing. Jiří Schimmel, PhD. [18].

Výsledný modul je sestaven jako dynamicky linkovaná knihovna, soubor typu `.dll`, a je nutné ho spouštět v hostitelské aplikaci, která zprostředkovává další komunikaci s ASIO ovladači. Pro debugování modulu byly používány hostitelské aplikace VST Analyzer a inTone, pro vyexportování výsledných signálů k následnému rozboru jejich spektrálních vlastností v prostředí MATLAB bylo použito DAW Cubase.

4.2 FFTW

Další knihovnou třetí strany použitou ve výsledném modulu je FFTW (Fastest Fourier Transform in the West), určená k výpočtu diskrétní Fourierovy transformace. Jejímí autory jsou zaměstnanci MIT, Matteo Frigo a Steven G. Johnson. Jejím specifikem oproti jiným knihovnám podobného zaměření je skutečnost, že před provedením samotné transformace je provedena inicializace objektu typu `plan`, který nese informace o optimálním postupu výpočtu, který vychází z konkrétních hardwarových možností počítače, na němž je knihovna právě používána. Další možností optimalizace časové a výpočetní náročnosti je ukládání vytvořených `plans` v objektech typu `wisdom`.

Pro účely práce byly použity plány pro jednorozměrnou diskrétní Fourierovu transformaci typu `r2c` pro dopřednou, a `c2r` pro zpětnou transformaci, které pracují pouze s polovinou spektra, komplexně sdružené vzorky vynechávají:

```
fftwf_plan fftwf_plan_dft_r2c_1d(int n0,
                                   float *in, fftwf_complex *out,
                                   unsigned flags);
fftwf_plan fftwf_plan_dft_c2r_1d(int n0,
                                   fftwf_complex *in, float *out,
                                   unsigned flags);
```

Datový typ `fftwf_complex` reprezentuje komplexní číslo v algebraickém tvaru a je reprezentováno dvouprvkovým polem datového typu `double`, kdy prvek s nulovým indexem nese hodnotu reálné části, prvek s indexem rovným jedné potom hodnotu imaginární. Parametr `int n0` reprezentuje fyzickou délku transformace, `*in` a `*out` jsou ukazatele na místa v paměti, ze kterých aplikace bere vstupní vzorky resp. do kterých zapisuje vzorky výstupní. Za argument `flags` byla zvolena hodnota `FFTW_ESTIMATE`, která zajistí vytvoření plánu rychleji než možnost `FFTW_MEASURE`.

Dále je také potřeba zmínit, že transformace typu `c2r` defaultně ničí svoje vstupní data, a to i pro *out-of-place* transformace. Pro algoritmy implementované v této práci to však nepředstavuje problém.

5 VÝSLEDKY PRÁCE

5.1 Struktura programu

Vytvořený modul vychází z projektu výše zmíněné šablony, do které byly kromě souborů knihovny FFTW přidány také soubory `CGenerator.cpp` a `CGenerator.h`, které představují vlastní processingové jádro aplikace. Výpis 5.1 je ukázkou hlavičkového souboru třídy `CGenerator` a ukazuje její nejdůležitější členské metody a proměnné.

Výpis 5.1: `CGenerator.h`.

```
class CGenerator {

    std::valarray<double> buffer;
    size_t numOfSamples;

    //funkce pro generování
    void whiteGen();
    void colorGen(enum generate);
    void MLSGen();
    void linSweepGen();
    void logSweepGen();
    void multitoneGen();

public:
    CGenerator();
    CGenerator(double sampleRate, int len);
    ~CGenerator();
    size_t samplePosition;

    struct TSign signalParameters;
    std::vector<harm> multitoneData;

    float getValue();
    void fillTheBuffer();
};
```

Tato třída zapouzdřuje v prvé řadě privátní proměnnou `valarray<double> buffer`,

která uchovává hodnoty vygenerované funkcemi pro generování jednotlivých typů signálů (tyto metody jsou blíže popsány u příslušných kapitol věnujících se rozboru výsledných signálů). Plugin tedy vzorky negeneruje v reálném čase, buffer je pouze dynamicky přegenerován po každém zavolání funkce `void fillTheBuffer()`.

Při každém volání funkce `parameterChanged(VstInt32 index, float value)` z vlastní třídy pluginu jsou aktuální hodnoty parametrů ukládány do struktury `TSign signalParameters`. Výjimku tvoří parametry *kPhmulti*, *kAmulti* a *kFmulti*, jejichž hodnoty jsou uchovávány ve vektoru struktur reprezentujících jednotlivé harmonické složky multitonu, `std::vector<harm> multitoneData`. Z těchto kontejnerů si bere funkce `fillTheBuffer()` parametry pro každé generování. Výpis 5.2 ukazuje definice použitých datových struktur.

Výpis B.1 ukazuje pole parametrů pluginu, rozsahů jejich reálných hodnot, způsobů jejich zobrazení atd.

Dále třída `CGenerator` zapouzdřuje funkci `float getValue()`, která vyčítá hodnoty odpovídající aktuální hodnotě proměnné `samplePosition` (která je s každým zavoláním této funkce inkrementována) z bufferu, převádí je na datový typ `float` a vrací je metodě `processReplacing()` třídy `CVSTTemplate`.

U signálů šumového typu funguje `getValue()` na principu čtení kruhového bufferu a po přehrání obsahu bufferu je proměnná `samplePosition` vynulována a čtení probíhá opět od začátku bufferu. Naproti tomu pro signály založené na harmonických funkcích funkce `getValue()` vrací jedinou periodu signálu, po jejím přehrání pak vrací nulové vzorky. Je tak zamezeno fázovým diskontinuitám, které by způsobovaly problém zejména u signálu složeného z harmonických složek.

5.2 Spuštění a ovládání aplikace

Pro spuštění modulu je nutné umístit soubor `TestGEN.dll` do adresáře, ze kterého host načítá VST pluginy. Některé aplikace umožňují dynamickou volbu tohoto adresáře a možnost „Rescan the plugins“ i po jeho spuštění, jinak je nutné plugin nainstalovat ještě před spuštěním hostitelské aplikace. Dále je nutné umístit soubor knihovny FFTW, `libfftw3f-3.dll` do adresáře se spustitelným souborem hostitelské aplikace.

Po zavolání konstruktoru pluginy se zobrazí jeho GUI, které zachycuje obrázek 5.1.

Pomocí ovládacího prvku `COptionMenu` s označením *Type* uživatel volí typ generovaného signálu. Prvky typu `CHorizontalSlider` slouží k nastavení parametrů zvoleného signálu. Dále je implementována možnost zadávat parametry v reálných hodnotách pomocí příslušných prvků typu `CTextEdit`.

Výpis 5.2: Datové struktury pluginy.

```
struct TSign{

    bool bypass;
    enum generate typ;
    size_t mask;
    double CF;
    double f1;
    double f2;
    double T;
    double vol;
    size_t numOfSines;
    size_t whichSine;
}

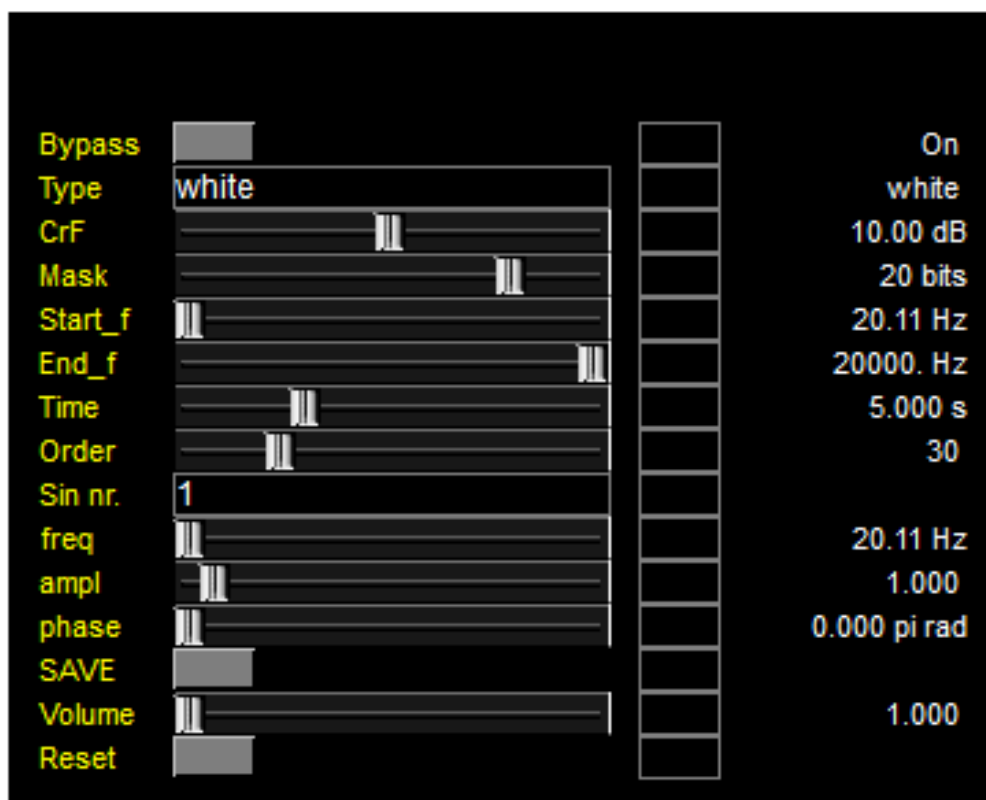
struct harm{
    double ampl;
    int freq;
    double phase;

    harm(){
        ampl = 1;
        freq = 0;
        phase = 0;
    };
}
```

Specifické je zadávání parametrů signálu složeného z harmonických složek. Pomocí slideru označeného *Order* se zadává řád multitonu, tedy počet harmonických složek. *COptionMenu Sin nr.* slouží k volbě indexu harmonické složky, pro kterou uživatel dále zadá příslušnými slidery frekvenci, amplitudu a počáteční fázi. Před zadáváním paramterů pro další harmonickou je nutné data uložit pomocí *CKickButton SAVE*.

Pro vygenerování signálu se zadanými parametry a jeho uložení do vnitřního bufferu slouží *CKickButton* s popisem *Reset*.

Ovládací prvky *Volume* a *Bypass* nepotřebují další popis.



Obr. 5.1: Grafické uživatelské rozhraní pluginu.

5.3 Rozbor výsledných signálů

5.3.1 Bílý šum

Prvním implementovaným signálem je bílý šum, který je generován funkcí `whiteGen()`. Algoritmus, na němž je založena, je již dostatečně popsán v teoretické části. Za zmínku dále stojí krok normalizace provedený před finálním zápisem prvků do bufferu, postup zachycuje výpis 5.3.

Výpis 5.3: Proces normalizace

```
/* Normalize */  
double absmax = std::abs(buffer).max();  
for (size_t i = 0; i < num0fSamples; i++)  
{  
    buffer[i] /= absmax;  
}
```

Grafy A.2 zobrazují Welchův odhad výkonové spektrální hustoty pro dvě různé zadané hodnoty činitele výkyvu. Pomocí přiloženého skriptu `colorNoiseAnalysis.m` byla pro tyto signály vypočítána skutečná hodnota $CF = 4,7932$ resp. $14,9093$ dB. Z grafů lze navíc dále vyčíst, že hodnota RMS odpovídá přibližně -5 resp. -15 dB, tedy reciproké hodnotě zadaného CF.

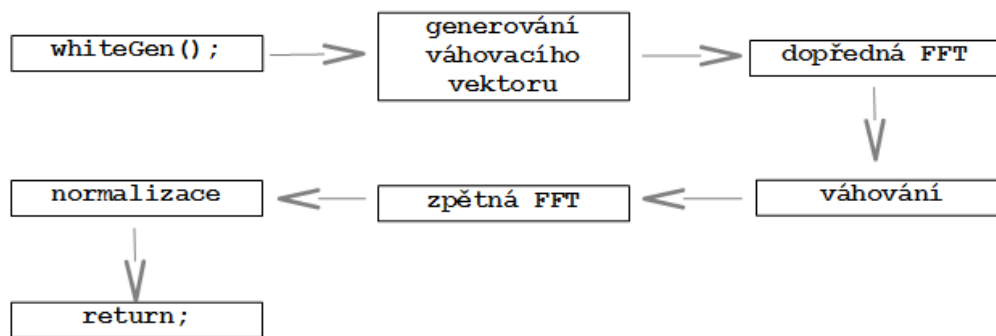
5.3.2 Barevné šumy

Generování šumů s modifikovaným frekvenčním spektrem zajišťuje v zásuvném modulu funkce `void colorGen(enum barvaSumu)`. Algoritmus znázorňuje blokové schéma 5.2.

Při zpětném převodu do časové oblasti způsobovala u růžového a červeného šumu problém vysoká stejnosměrná složka, proto je vždy před zavoláním funkce pro IFFT vynulována.

Z přiložených grafů A.3 a A.4 je dobře patrný sklon výkonové spektrální hustoty. Postup výpočtu jejího odhadu naznačuje ukázka skriptu `colorNoiseAnalysis.m` 5.4. Kreslicí procedura byla převzata z [21]).

Přiložené grafy jsou vytvořeny ze signálů generovaných při přemostění mapovací funkce pro úpravu činitele výkyvu bílého šumu. Dalším testováním však bylo ověřeno, že tato úprava nemá vliv na spektrální obsah výsledných barevných šumů, proto program umožňuje generování šumů s různým CF, jeho skutečná hodnota



Obr. 5.2: Algoritmus generování barevných šumů.

však obvykle neodpovídá uživatelem zadané hodnotě parametru, hodnoty se přibližně shodují jen na určitém intervalu. Například pro růžový šum tento je tento interval od cca 11,5 do 15 dB. Růžový šum s nižším CF než 11 dB se nepodařilo vygenerovat. Pro strmější sklony spektra jsou tyto intervaly shody ještě užší.

Výpis 5.4: Kreslicí procedura pro zobrazení spektra barevných šumů.

```

N = length(x);
fs = 44100;

winlen = fs/2;
window = hanning(winlen, 'periodic');
noverlap = winlen/2;
[Pxx,f] = pwelch(x, window, noverlap, [], fs, 'onesided');
PxxdB = 10*log10(Pxx);

```

5.3.3 Pseudonáhodný binární šum

Ve výsledném modulu je pseudonáhodný binární šum generován funkcí `MLSGen()`, která simuluje hardwarový způsob generování MLS. Posuvný registr je implementován jako kontejner typu `std::valarray<bool>`. Dále zde figuruje pomocná proměnná `buffer`, do níž je ukládán výsledek postupné XOR operace nad všemi prvky registru odpovídajícím nenulovým indexům použité masky (tzv. kohoutkům). Po bitovém posunutí registru je pak hodnota `bufferu` přiřazena na pozici prvního bitu.

V programu je implementováno generování s použitím masek o délkách 16 až 21 bitů. Pomocí skriptu `MLS_Analysis.m` bylo ověřeno splnění výše popsanych kritérií

pro všechny použité masky. Graf A.1 ukazuje autokorelační funkci jedné periody signálu MLS s 16bitovou maskou.

Oproti generování v prostředí MATLAB v semestrálním projektu musely být původní skripty upraveny, protože (patrně při renderingu testovacího signálu z DAW) došlo k určité chybě a ne všechny vzorky byly rovny přesně hodnotám -1 a 1 . Absolutní hodnota odchylky však nepřesáhla hodnotu $0,003$ jednotky.

5.3.4 Přeladovaný harmonický signál

Pro testování přeladovaných harmonických signálů byl vytvořen matlabovský skript `sweepAnalysis.m`. Z grafů A.5 je dobře patrný příslušný sklon spektra popsany v teoretické části práce. Při delších dobách přeladění (okolo 15 s) jsou na vysokých kmitočtech patrné určité spektrální artefakty. Pro většinu měřících konfigurací by však neměly představovat větší problém.

5.3.5 Signál složený z harmonických složek

Pro ověření správnosti generování multiharmonického signálu byl vygenerován vzorek odpovídající výše popsaným normám - dvousložkový signál pro pro měření intermodulačního zkreslení podle DIN a multiton obsahující 30 složek rovnoměrně rozložených v logaritmickém frekvenčním spektru. Výsledky vygenerované skriptem `multitoneAnalysis.m` zachycuje graf A.6.

6 ZÁVĚR

Předkládaná bakalářská práce se věnovala tematice softwarového generování testovacích signálů pro různé druhy akustických měření. Byl vytvořen funkční VST plugin, schopný generovat v praxi nejčastěji používané testovací signály s možností měnit jejich parametry. Pro usnadnění práce s modulem by bylo vhodné při dalším vývoji implementovat možnost ukládat různá nastavení parametrů zejména pro signály složené z harmonických složek, případně možnost náhodného nastavení fází jednotlivých složek. I přes tuto nedokonalost v uživatelském rozhraní je však plugin v praxi dobře použitelný, což dokazuje i výše uvedený rozbor vygenerovaných vzorků signálů.

LITERATURA

- [1] BOULANGER, Richard Charles a Victor LAZZARINI. *The audio programming book*. Cambridge, Mass.: MIT Press, 2011, xxiii, 889 p. ISBN 0262014467.
- [2] CHAN, I.H. *Swept Sine Chirps for Measuring Impulse Response*. Stanford Research Systems, 2010, 6 s. [cit. 2015-12-08]. Dostupné z URL: http://www.thinksrs.com/downloads/PDFs/ApplicationNotes/SR1_SweptSine.pdf
- [3] CHITODE, J.S. *Principles of Communication*. Technical Publications, Pune, India, 2009, First Edition, 559 s. ISBN 978-81-8431-661-2
- [4] COOLEY, James W. a John W. TUKEY. *An Algorithm for the Machine Calculation of Complex Fourier Series*. DOI: 10.2307/2003354. ISBN 10.2307/2003354. Dostupné také z: <http://www.jstor.org/stable/2003354?origin=crossref>
- [5] HAVELOCK, D.I., KUWANO S., VORLANDER, M. *Handbook of signal processing in acoustics*. New York, NY: Springer, c2008, 2 v. (xxv, 1950 p.). ISBN 9780387304410-0.
- [6] HERLES V. *Amplitudové spektrum binárních, zejména pseudonahodných signálů*. Zpráva ÚTIA č. 195, září 1967.
- [7] KASDIN, N.J., T. WALTER. *Discrete simulation of power law noise (for oscillator stability evaluation)*. In: Proceedings of the 1992 IEEE Frequency Control Symposium [online]. IEEE, 1992, 2015-11-19, s. 274-283 [cit. 2015-11-19]. DOI: 10.1109/FREQ.1992.270003. ISBN 0-7803-0476-4. Dostupné z URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=270003>
- [8] van den HEUVEL, B., van den BRINK, R. *PSD + Crest factor is not sufficient to specify noise in performance tests*. Project VDSL, part 1, 6 s. [cit. 2015-12-07]. Dostupné z URL: http://www.joepeesoft.com/Public/DSL_Corner/Docs_ETSI_TM6/983t16a0.pdf.
- [9] HEYSER, R. C. *Acoustical Measurements by Time Delay Spectrometry*. In: Journal of Audio Engineering Society, Vol.15 , Issue 4, 1967.
- [10] van den HEUVEL, B., van den BRINK, R. *Specification of crest distribution mask for noise in performance tests*. Project VDSL, part 1 , 6.s. [cit. 2015-12-07]. Dostupné z URL: http://www.joepeesoft.com/Public/DSL_Corner/Docs_ETSI_TM6/984t42a0.pdf.

- [11] MASSARANI, P., MÜLLER S. *Transfer-Function Measurement with Sweeps*. In: Journal of Audio Engineering Society, Vol. 49, Issue 6, str.443-471. červen 2001 [online]. [cit. 2015-12-07]. Dostupné z URL: <http://www.aes.org/e-lib/browse.cfm?elib=10189>
- [12] METZLER, B. *Audio Measurement Handbook*. Beaverton, US-OR, Audio PRECISION, Inc., 2005, 2. v., 171 s. ISBN 978-9994569885.
- [13] Martínez Núñez, L.R. *Procesamiento Digital y Control Gestural en Tiempo Real Utilizando una PC con Drivers ASIO para Efectos de Audio*. Puebla, México, 2007. Tesis Licenciatura. Ingeniería en Electrónica y Computadoras. Departamento de Computación, Electrónica, Física e Innovación, Escuela de Ingeniería y Ciencias, Universidad de las Américas Puebla.
- [14] MARSAGLIA, G., TSANG W. W. *The Ziggurat Method for Generating Random Variables*. Journal of Statistical Software [online]. 2000, 2015-11-23, 5(8): - [cit. 2015-11-23]. DOI: 10.18637/jss.v005.i08. ISSN 1548-7660. [cit. 2015-12-07]. Dostupné z URL:<http://www.jstatsoft.org/v05/i08/>
- [15] MÜLLER, G. (ed.). *Handbook of engineering acoustics*. Berlin: Springer, c2013, x, 702 s. ISBN 978-3-540-24052-5.
- [16] SOPER, Peter. *Calibrating Cinema Sound Systems*. Meyer Sound Laboratories Inc. 2012. 4 s. Technical Report. [cit. 2015-12-07]. Dostupné z URL: http://www.meyersound.com/pdf/cinema_technical_papers/cinema_calibration_tech_report.pdf.
- [17] SCHIMMEL, J. *Elektroakustika*. Elektronická skripta, VUT v Brně, 2014.
- [18] SCHIMMEL, Jiří. *Šablona VST plug-in modulu*. Verze 1.2. 4.s. VUT v Brně, 2008.
- [19] TŮMA, Jiří. *Zpracování signálů získaných z mechanických systémů užitím FFT*. Praha: Sdělovací technika, 1997, vii, 174 s. ISBN 80-901936-1-7.
- [20] SMÉKAL, Z. *Analýza signálů a soutav - BASS*. Elektronická skripta, VUT v Brně, 2012.
- [21] ZHIVOMIROV, H. *Arbitrary spectral slope noise generation with Matlab implementation*. [cit. 2016-05-29]. Dostupné z URL: <https://www.mathworks.com/matlabcentral/fileexchange/48628-arbitrary-spectral-slope-noise-generation-with-matlab-implementation>

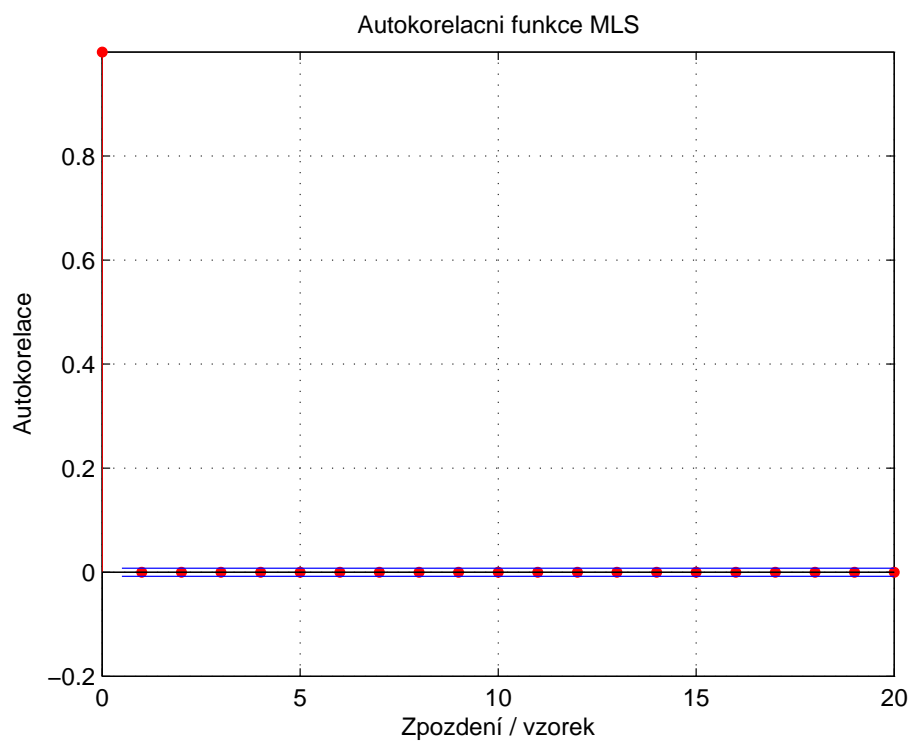
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ASIO	Audio Stream Input Output
CCIF	Comité consultatif international téléphonique
CF	crest factor
DAW	Digital Audio Workstation
DIN	Deutsches Institut für Normung, e.V.
FFT	Fast Fourier Transform, rychlá Fourierova transformace
FFTW	Fastest Fourier Transform in the West, nejrychlejší FFT na západě
FHT	Fast Hadamard Transform, rychlá Hadamardova transformace
f_{vz}	vzorkovací kmitočet
GUI	Graphic User Interface, grafické uživatelské rozhraní
IFFT	Inverse Fast Fourier Transform, inverzní rychlá Fourierova transformace
MIT	Massachusetts Institute of Technology, Massachusettský technologický institut
MLS	Maximum Length Sequence, posloupnost maximální délky
PSD	Power Spectral Density, výkonová spektrální hustota
RMS	Root Mean Square, efektivní hodnota
SDK	Software Development Kit, systémový vývojový nástroj
SMPTE	Society of Motion Picture and Television Engineers
SNR	Sound to Noise Ratio, odstup signálu od šumu
TDS	Time Delay Spectrometry
THD	Total Harmonic Distortion, celkové harmonické zkreslení
VST	Virtual Studio Technology

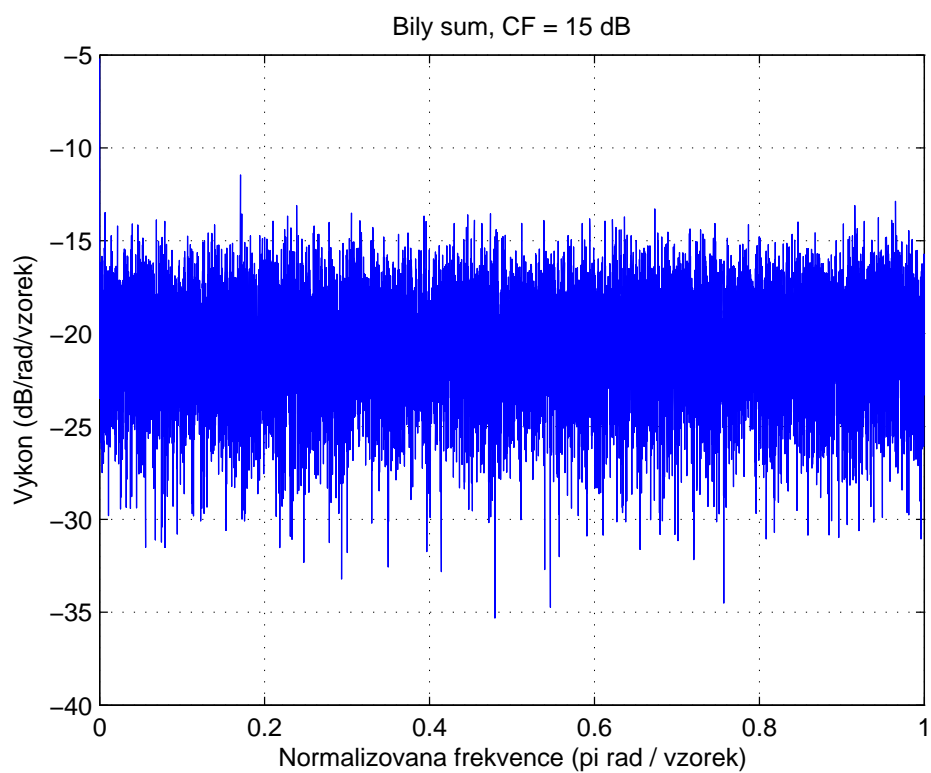
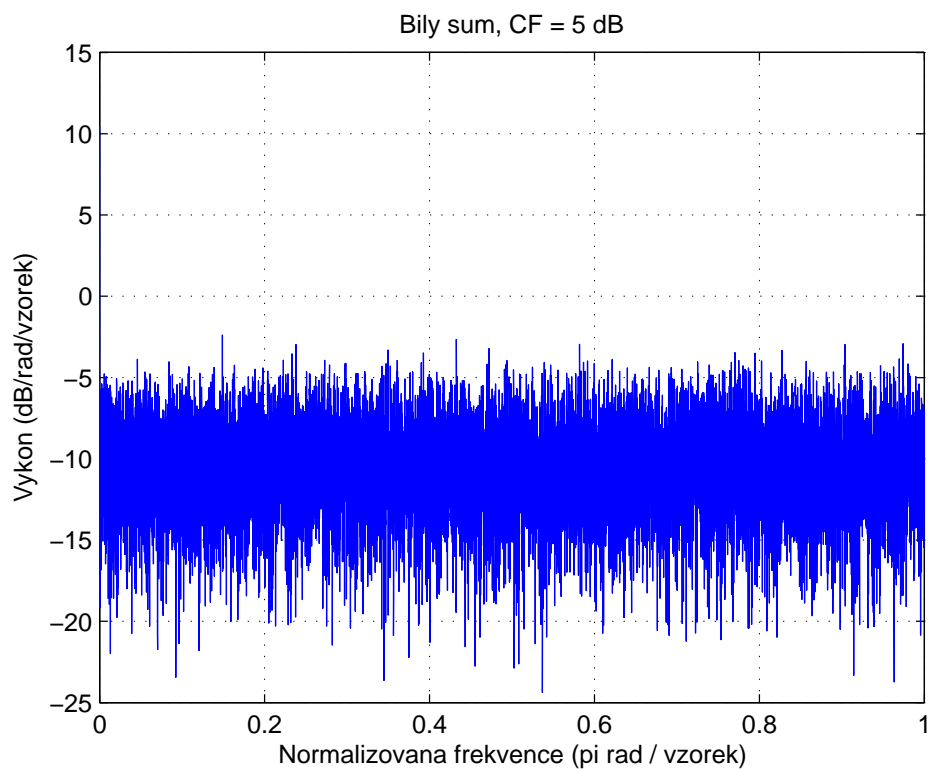
SEZNAM PŘÍLOH

A	Spektrální vlastnosti výsledných signálů	39
B	Ukázky kódu	45
C	Obsah přiloženého DVD	47

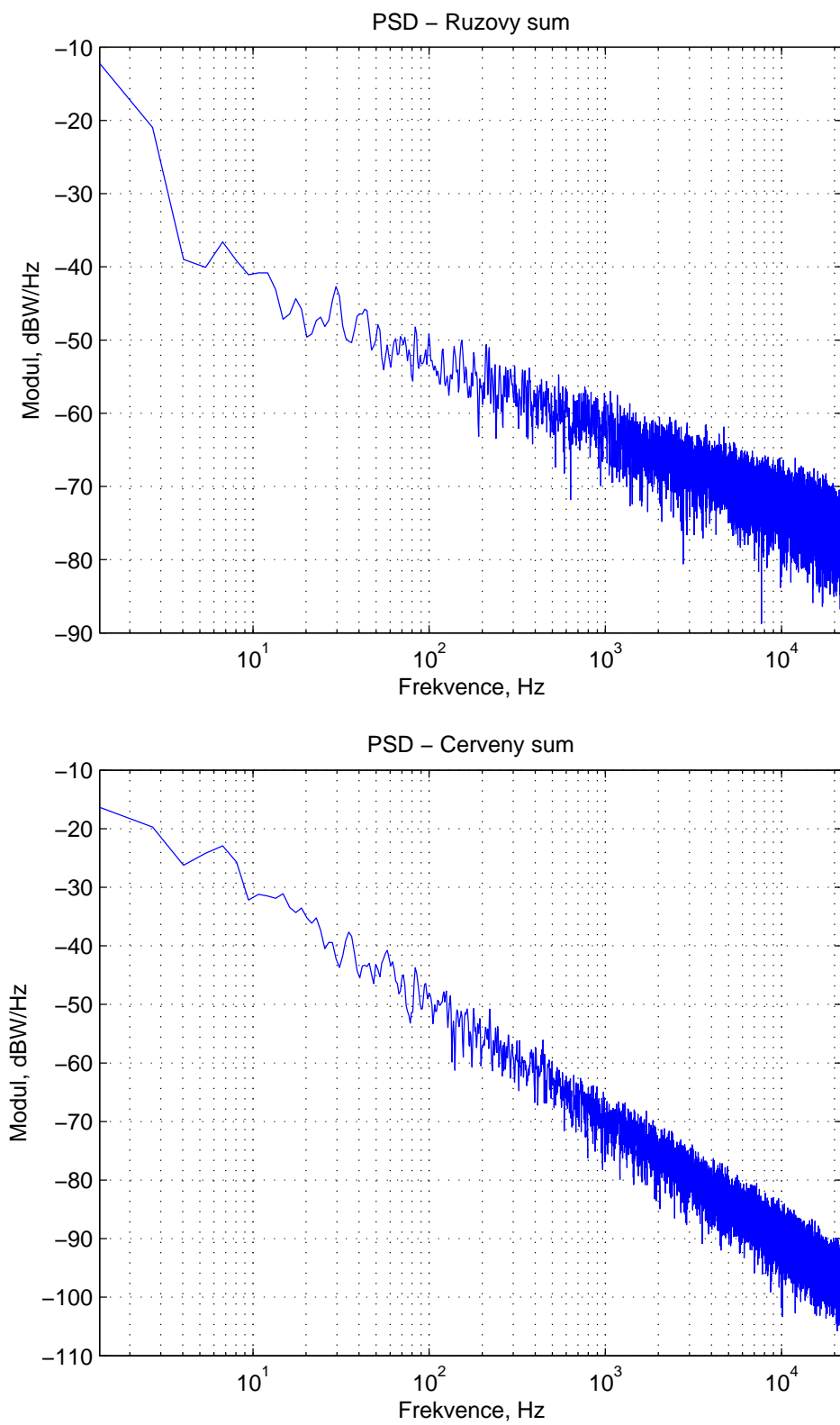
A SPEKTRÁLNÍ VLASTNOSTI VÝSLEDNÝCH SIGNÁLŮ



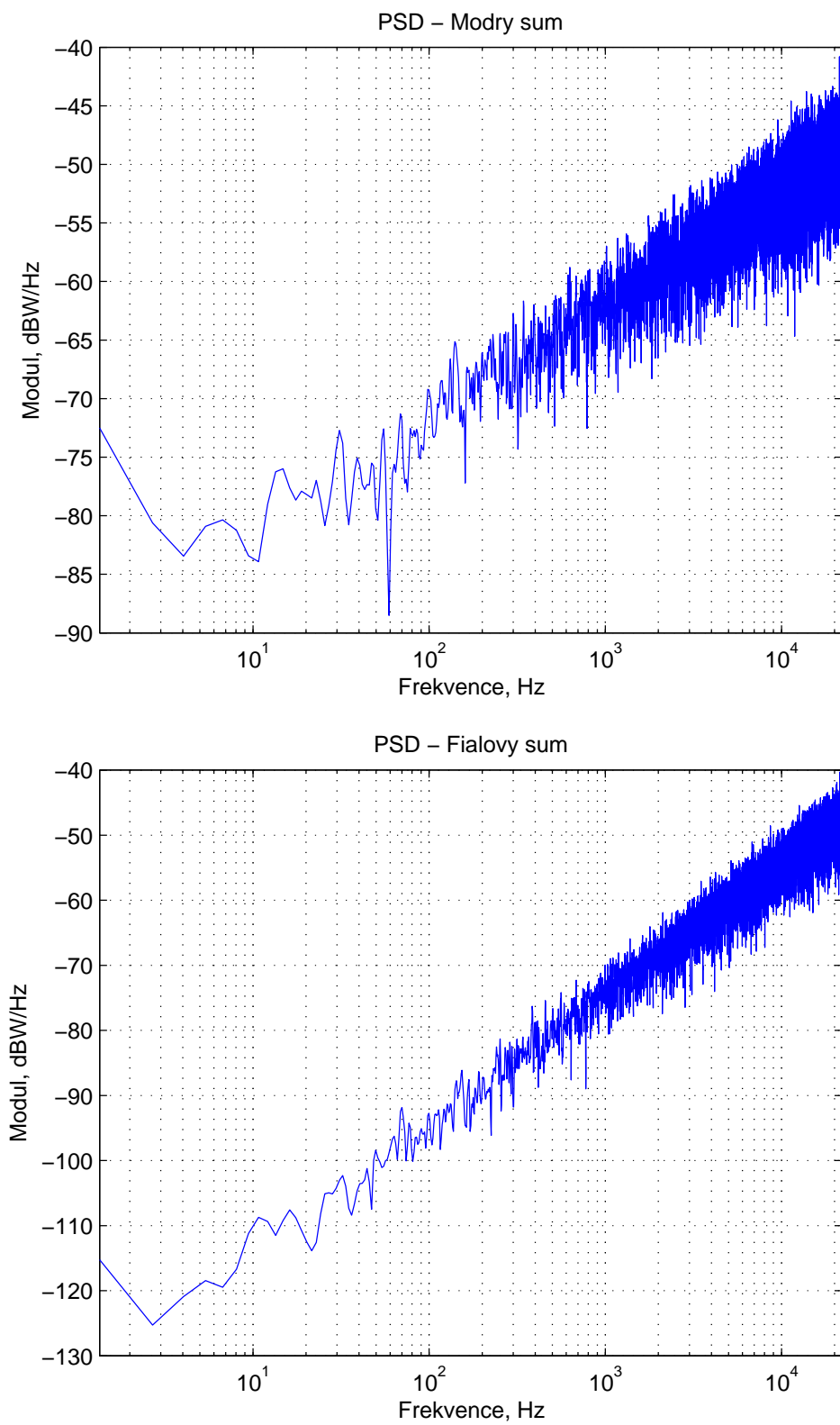
Obr. A.1: Autokorelační funkce MLS s délkou masky 16 bitů.



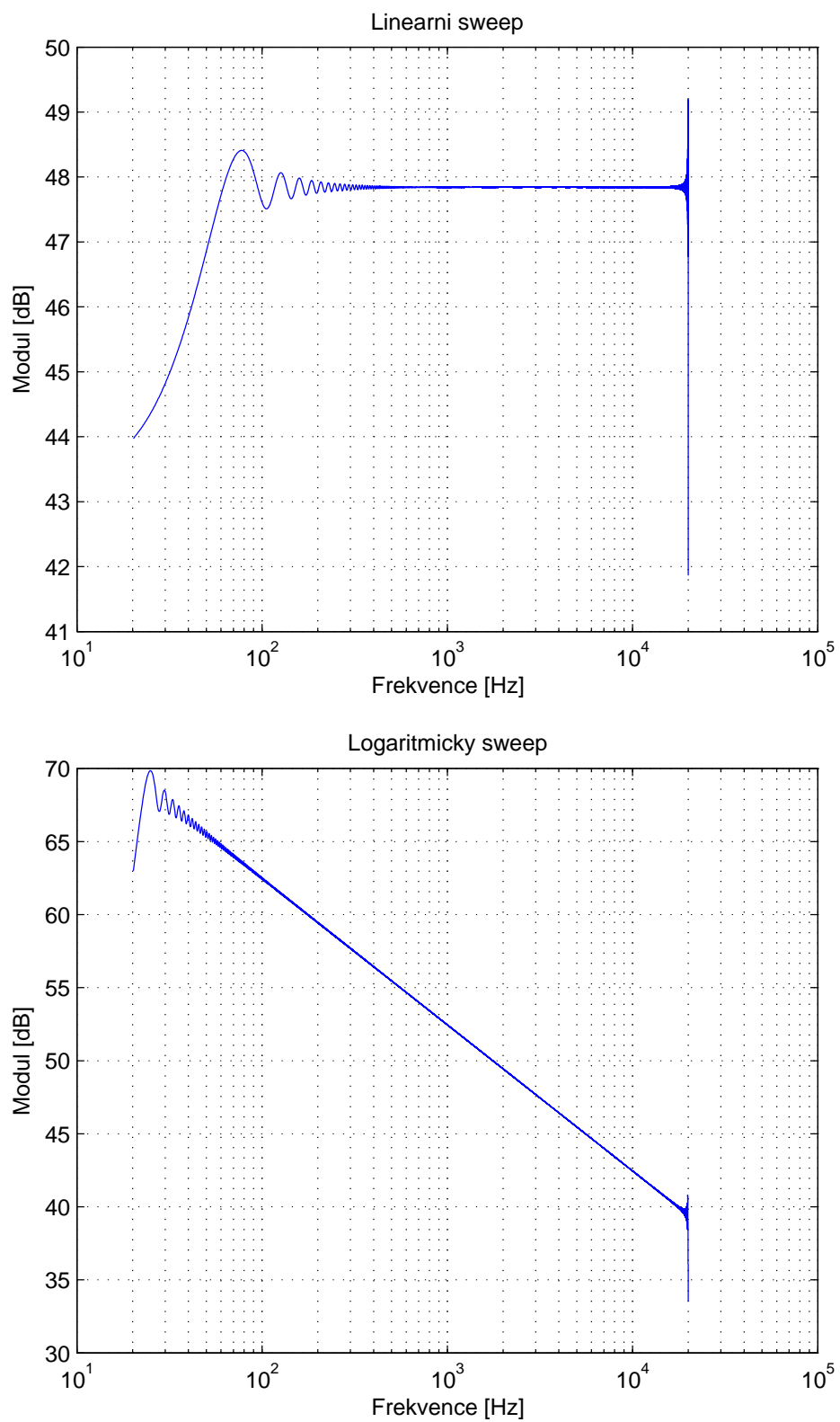
Obr. A.2: PSD bílého šumu s různými hodnotami CF.



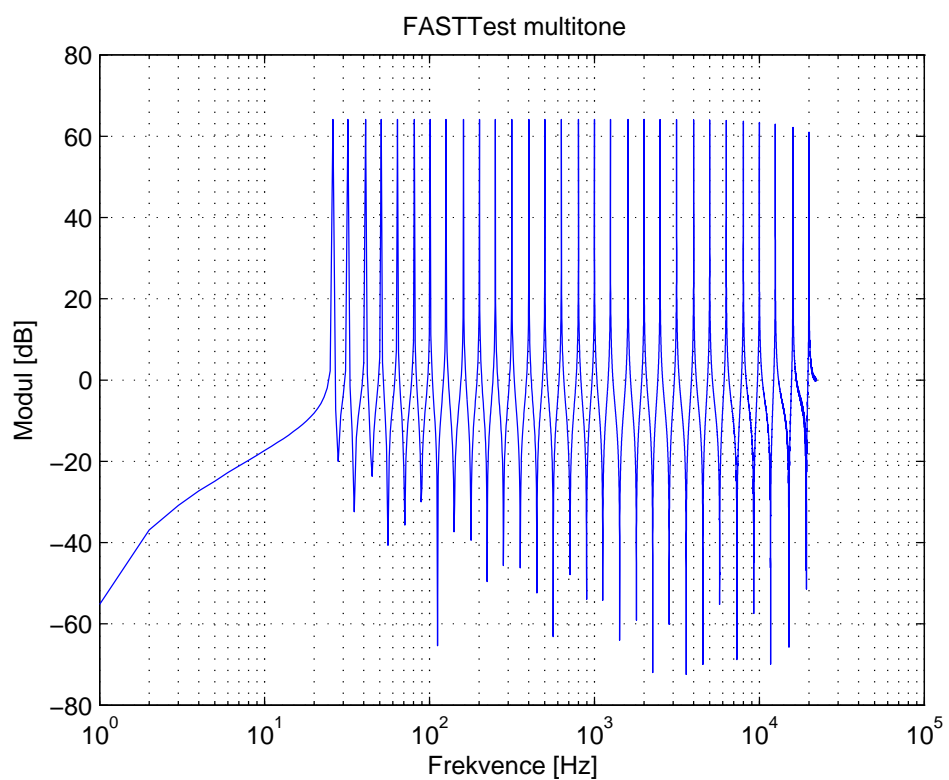
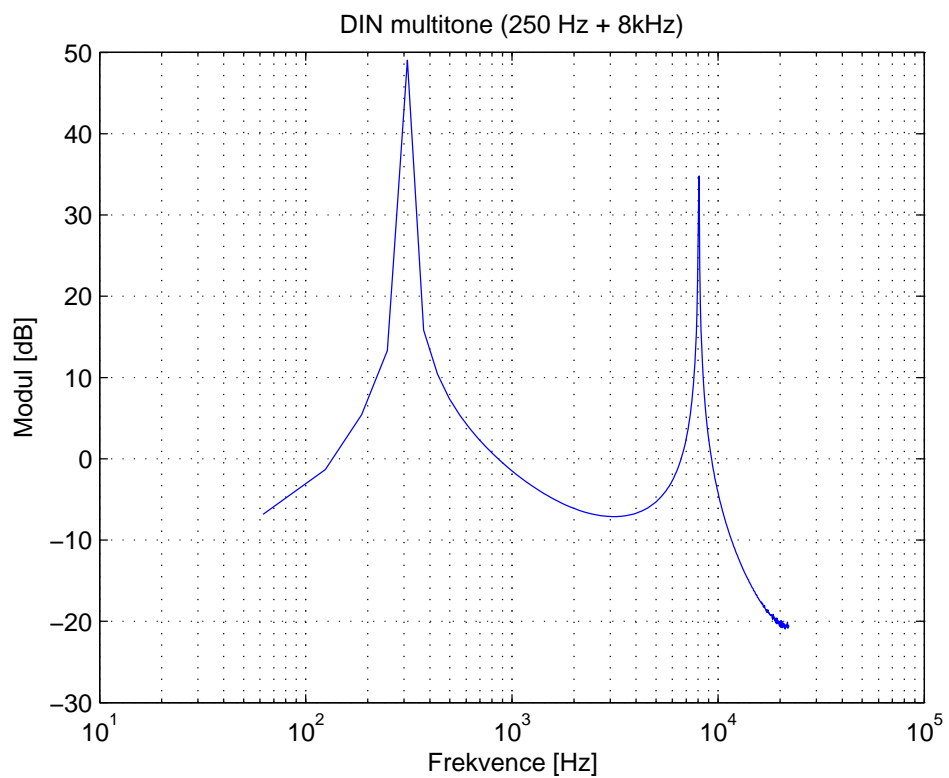
Obr. A.3: Spektrální vlastnosti růžového a červeného šumu.



Obr. A.4: Spektrální vlastnosti modrého a fialového šumu.



Obr. A.5: Spektrální vlastnosti přeladovaných harmonických signálů.



Obr. A.6: Spektrální vlastnosti signálů složených z harmonických složek.

B UKÁZKY KÓDU

Výpis B.1: Struktura parametrů.

```
ParamInfo unsortedParams[PARAM_NUMBER] =
{
/* tag, current, default, min, max, name, label, show,
   read only */
{kBypass, 0.0f, 0.0f, 0.0f, 1.0f, "Bypass", "", SHOW_ONOFF,
false},
{kTypeOfSignal, 1.0f, 1.0f, 1.0f, 9.f, "Type","", SHOW_NAME,
false},

//šumy
{kCF, 10.0f, 10.0f, 5.f, 15.0f, "CrF", "dB", SHOW_FLOAT,
false},

//MLS
{kMLSmask, 20.f,20.0f, 16.0f, 21.0f, "Mask", "bits", SHOW_INTEG,
false},

//sweep
{kf1, 20.0f, 20.f, 16.f, 20000.0f, "Start_f", "Hz", SHOW_LOG,
false},
{kf2, 20000.0f, 20000.0f, 16.0f, 20000.0f, "End_f", "Hz", SHOW_LOG,
false},
{kT, 5.0f, 5.0f, 1.0f, 15.0f, "Time", "s", SHOW_FLOAT,
false},

//multitone
{kNumOfSines, 30.0f, 30.0f, 1.0f, 128.0f, "Order", "",SHOW_INTEG,
false},
{kWhichSine, 1.0f, 1.0f, 1.0f, 128.f, "Sin_nr.", "",SHOW_NR,
false},
{kFmulti, 20.0f, 20.f, 16.f, 20000.0f, "freq", "Hz", SHOW_LOG,
false},
{kAmulti, 1.0f, 1.0f, 0.0f, 15.0f , "ampl", "", SHOW_FLOAT,
false},
{kPhmulti, 0.0f, 0.0f, 0.0f, 2.0f, "phase", "pi_rad", SHOW_FLOAT,
false},
{kMultiSetup, 0.0f, 0.0f, 0.0f, 1.0f, "SAVE", "", SHOW_TRIGGER,
false},
```

```
{kVolume, 1.0f, 1.0f, 1.0f, 15.0f, "Volume", "", SHOW_FLOAT,
false},

//setup parameters
{kParSetup, 0.0f, 0.0f, 0.0f, 1.0f, "Reset", "", SHOW_TRIGGER,
false}

};
```

C OBSAH PŘÍLOŽENÉHO DVD

Příloha obsahuje zdrojové soubory výsledného zásuvného modulu, vyrenderované zvukové ukázky testovacích signálů, skripty vytvořené v programu MATLAB, verze R2013a, určené k rozboru jejich spektrálních vlastností a text práce ve formátu PDF.